IN THE CLAIMS:

Please amend the claims as follows:

1. (Currently Amended) A pipeline, comprising:

a plurality of operational stages, the stages including:

a pointer register stage which stores pointer information and updates;

a pointer dependency checking stage located downstream of the pointer register stage, which determines if instruction pointer dependencies exist and stalls an instruction ~~issue~~ prior to issuance if necessary to resolve inter-instruction dependencies;

at least one functional unit providing pointer information updates to the pointer register stage such that the pointer information is processed and updated to the pointer register stage before an instruction goes through the dependency checking stage, an issue stage, and an execution stage.

2. (Original) The pipeline as recited in claim 1, further comprising a pointer execution stage used before the dependency checking stage such that inter-instruction dependency is checked after the pointer execution stage or in parallel with pointer execution.

3. (Original) The pipeline as recited in claim 2, further comprising a path for making pointer updates available to the pointer register stage before the instruction reaches a write back stage of the pipeline.

4. (Original) The pipeline as recited in claim 3, wherein the path includes a normal pointer update path which returns pointer information from the at least one functional unit.

5. (Original) The pipeline as recited in claim 3, further comprising a pointer execution bypass making pointer updates available to immediately following instructions before a pointer update is written into the pointer register file.

6. (Original) The pipeline as recited in claim 2, further comprising a pointer reorder buffer coupled to the pointer register stage to maintain a precise state of pointers.

7. (Original) The pipeline as recited in claim 6, further comprising a precise pointer file for storing the precise state of the pointer reorder buffer.

8. (Original) The pipeline as recited in claim 7, further comprising an interrupt recovery path which keeps the pointer register stage up to date with reordering or recovery information from the precise pointer file.

9. (Original) The pipeline as recited in claim 1, further comprising a combined pointer reorder buffer/issue stage coupled to the dependence checking stage to issue instructions and maintain a precise state/order of pointers.

10. (Original) The pipeline as recited in claim 9, wherein the combined pointer reorder buffer/issue stage includes a table with a plurality of fields to keep identifiers of all pointers that are updated by an instruction, and new values of the updated pointers.

11. (Original) The pipeline as recited in claim 9, further comprising a precise pointer file for storing the precise state of the combined pointer reorder buffer/issue stage.

12. (Original) The pipeline as recited in claim 11, further comprising an interrupt recovery path, which restores the pointer register stage to the precise state from the precise pointer file.

13. (Previously Presented) A pipeline, comprising:

a plurality of operational stages, the stages including:

a pointer register stage which stores pointer information and updates;

a dependence checking stage located downstream of the pointer register stage, which determines if instruction dependencies exist and stalls an issue prior to issuance if necessary to resolve inter-instruction dependencies;

a pointer execution stage for processing pointers prior to the dependence checking stage, the pointer execution stage providing pointer updates to the pointer register stage via an early pointer update path; and

at least one functional unit providing pointer information updates to the pointer register stage such that pointer information is processed and updated to the pointer register stage.

14. (Original) The pipeline as recited in claim 13, further comprising a pointer reorder buffer coupled to the pointer register stage to maintain a precise state of pointers.

15. (Original) The pipeline as recited in claim 14, further comprising a precise pointer file for storing the precise state of the pointer reorder buffer.

16. (Original) The pipeline as recited in claim 15, further comprising an interrupt recovery path which keeps the pointer register stage up to date with reordering or recovery information from the precise pointer file.

17. (Original) The pipeline as recited in claim 13, further comprising a normal pointer update path which returns pointer information from the at least one functional unit.

18. (Original) The pipeline as recited in claim 13, further comprising a combined pointer reorder buffer/issue stage coupled to the dependence checking stage to issue instructions and maintain a precise state/order of pointers.

19. (Original) The pipeline as recited in claim 18, wherein the combined pointer reorder buffer/issue stage includes a table with a plurality of fields to keep identifiers of all pointers that are updated by an instruction, and new values of the updated pointers.

20. (Original) The pipeline as recited in claim 18, further comprising a precise pointer file for storing the precise state of the combined pointer reorder buffer/issue stage.

21. (Original) The pipeline as recited in claim 20, further comprising an interrupt recovery path, which restores the pointer register stage to the precise state from the precise

pointer file.

22. (Currently Amended) A method for updating pointers ahead of an instruction, comprising the steps of:

providing a plurality of operational stages, including a pointer register stage which stores pointer information and updates, a pointer dependence checking stage located downstream of the pointer register stage, which determines if instruction dependencies exist and stalls an issue prior to issuance if necessary to resolve inter-instruction dependencies, and at least one functional unit providing pointer information updates to the pointer register stage; and

processing pointer information to update the pointer information for the pointer register stage so that updated pointer information is available such that the pointer information is processed and updated to the pointer register stage before an instruction goes through the dependency checking stage, an issue stage, and an execution stage.

23. (Original) The method as recited in claim 22, further comprising a step of providing pointer updates to the pointer register stage via an early pointer update path by providing a pointer execution stage used before the pointer register stage and the dependence checking stage.

24. (Original) The method as recited in claim 23, further comprising a step of maintaining a precise state of pointers by employing a pointer reorder buffer.

25. (Original) The method as recited in claim 24, further comprising a step of storing the

precise state of the pointer reorder buffer in a precise pointer file.

26. (Original) The method as recited in claim 24, further comprising updating reordering or recovery information from the precise pointer file using an interrupt recovery path to the pointer register stage.

27. (Original) The method as recited in claim 23, further comprising maintaining a precise state/order of pointers using a combined pointer reorder buffer/issue stage coupled to the rename and dependence checking stage.

28. (Original) The method as recited in claim 27, wherein the combined pointer reorder buffer/issue stage includes a table with a plurality of fields to keep identifiers of all pointers that are updated by an instruction, and new values of the updated pointers.

29. (Original) The method as recited in claim 27, further comprising storing the precise state of the combined pointer reorder buffer/issue stage using a precise pointer file.

30. (Original) The method as recited in claim 29, further comprising an interrupt recovery path which keeps the pointer register stage up to date with reordering or recovery information from the precise pointer file.